InverseFilterbank Validation

We've identified three algorithm elements that will hopefully contribute to improving PFB inversion performance. I'll discuss each of these factors individually.

- PFB sample alignment
- Deripplingcorrection
- Overlap discard regions for streaming data

We've also switched to using a different PFB channelization implementation that matches the one used by Ian Morrison in his paper discussing FFT based PFB inversion. This implementation differs from the older PFB channelization implementation in that it pads input data differently, and uses a forward FFT instead of an inverse FFT. Moreover, it peforms roughly twice as well as the old algorithm.

In the end, I'll talk about the current state of FFT PFB inversion performance, in both the temporal and spectral regimes in the context of SKA performance specifications.

Error analysis

We're using two quantities to help assess the purity of the PFB inversion algorithm: max spurious power, total spurious power and mean spurious power. In the context of the PST specification, only the max and total spurious power are needed. The spurious power of an array is where the maximum value of the array is set to zero. In Python:

```
import numpy as np

def spurious(a):
  a[np.argmax(a)] = 0.0
  return a
```

In the case of analyzing time domain impulses, we're concerned with calculating these quantities with respect to the time domain signal. In the case of looking at complex sinuosoids, we're interested in calculating these values with respect to the frequency domain signal.

Cross correlation

In the section on sample alignment, I talk about the cross correlation between two signals. In reality, I'm not talking about the true cross correlation, defined (in terms of the convolution theorem) as follows:

$$f \star g = \mathcal{F}^{-1}\{\mathcal{F}\{f\} * \overline{\mathcal{F}\{g\}}\}$$

Rather, I'm talking about the term before the inverse fourier transform:

$$\mathcal{F}\{f\} * \overline{\mathcal{F}\{g\}}$$

This quantity is useful because the phase will be zero if the two signals are aligned in time. In principle, the lag, or the magnitude of the true cross correlation tells us the time offset between two signals, but the phase can be more illustrative in describing *how* two signals are out of phase.

PFB sample alignment

Care must be taken when choosing at which sample to start inverting oversampled channelized data. Take, for example, a signal that has been split into 16 channels with an 8/7 oversampling ratio. The PFB analysis algorithm intakes 14 samples for every one channelized sample. This, in turn, corresponds to 14 samples in the single channel result of the PFB synthesis algorithm. If we were to skip the first sample of channelized data, we would introduce a phase shift between the original input data and the inverted data. This phase shift cannot be correctly simply by offsetting the input or inverted output an expected number of samples – it is inherent in the inverted data due to our choice of FFT window alignment.

Each of the reported plots contains five subplots. Each of these subplots shows the following.

1. The input data, a time domain impulse. The impulse is a single bin wide, and is placed somewhere near the middle of the block.
2. The input data in log space.
3. The result of PFB inversion.
4. The result of PFB inversion in log space. This makes it easier to see any temporal leakage.
5. The phase of the fft cross correlation between input and inverted time series.

For each of the reported plots, I use a 80 tap filter with derippling enabled. The PFB channelizer creates 8 channels, with 4/3 oversampling ratio. The size of the forward FFT used in PFB inversion is 1024 samples. We're looking at a single block of data, in order to ignore any effects that might be introduced due to block-based processing. Using a smaller number of channels makes it easier to see the phase of the cross correlation.

Figure 1 shows the result of properly aligned PFB inversion. The phase of the cross correlation (in the fourth subplot) is not exactly zero, but it is close.

Figure 2 shows the effect of introducing a single sample shift in the channelized data. This shift introduces a phase gradient in the cross correlaton.

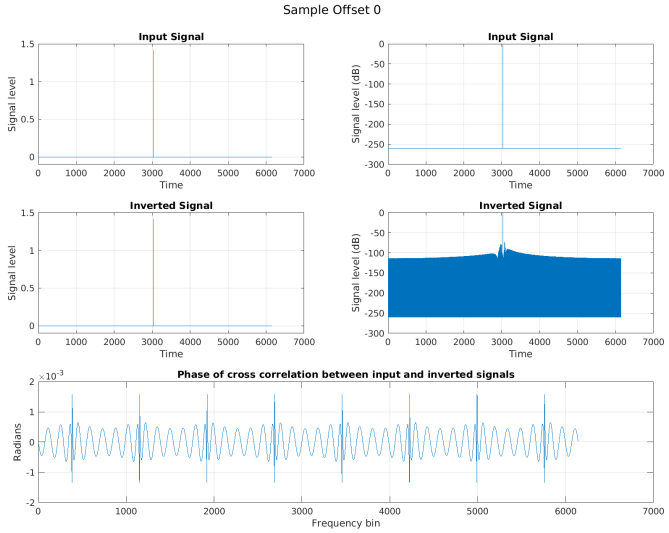

Figure 1: No sample offset

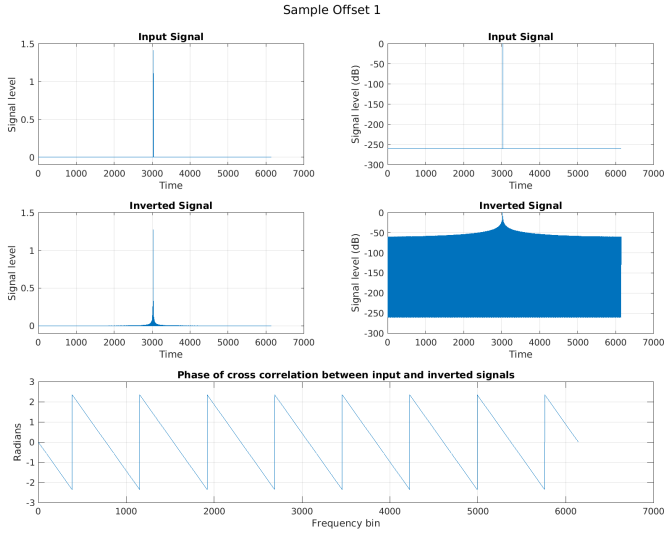

Figure 2: Single sample offset. Notice greater temporal leakage

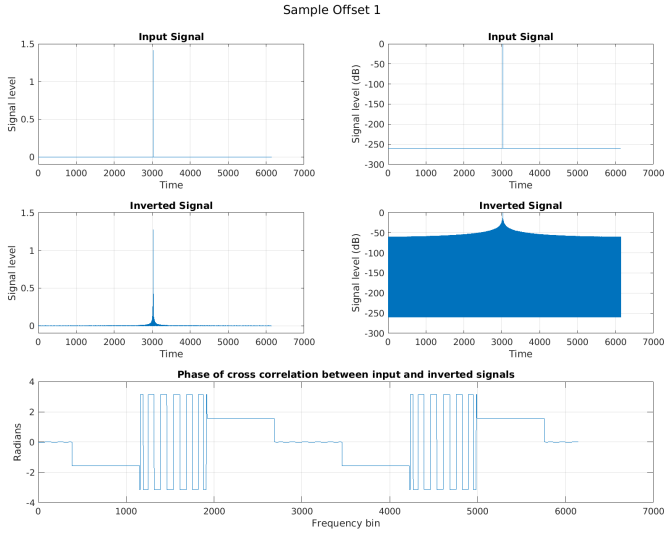

Figure 3: Single sample offset with correction

Figure 3 shows the effect of attempting to correct for the single sample shift. In this case, we introduce a seven sample shift in the input data (equal to the denominator of the oversampling ratio multiplied times the sample shift). This flattens the phase response in each subband (corresponding to the boundaries of forward FFTs in PFB inversion), but does not flatten the response across the band.

Figure 4 shows the effect of introducing an eight sample shift in the channelized data. Given that this is an integer multiple of the numerator of the oversampling ratio, the phase response is again flat.

Only by aligning the PFB inversion's FFT window to an integer multiple of the numerator of the oversampling ration can we correctly align the PFB inversion algorithm in phase, thereby significantly reducing temporal leakage.

Filter design

The PFB inversion phase alignment problem has implications for FIR filter design. We must choose a number of filter taps such that the following condition is satisfied:

$$\frac{N_{taps} - 1}{2N_{chan}} \in \mathbb{Z}$$

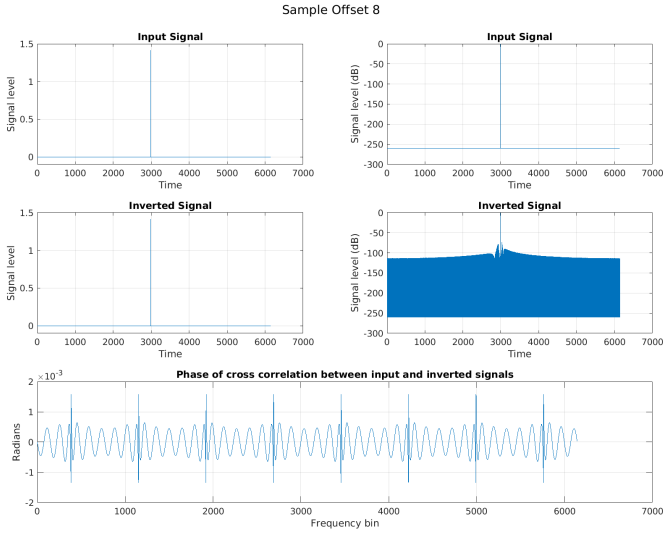

Figure 4: Eight sample offset

Here, the number of filter taps is always odd. If this condition is not satisfied, channelization introduces an inherit non zero sample shift to the data.

Derippling correction

Derippling means correcting for a FIR filter's passband. For a relatively small number of taps, the passband can have a significant amount of ripple.

Figure 5 shows the filter response of a FIR filter with 64+1 taps. In practice, a FIR filter would have significantly more taps, as finer channelization is often desired. In the context of PFB inversion, derippling involves dividing the result of each forward FFT with the passband region of the FIR filter frequency domain response.

For all the plots in this section, I use a 256 channel PFB, with 4/3 oversampling.

Derippling and temporal purity

Figure 6 shows how applying the derippling correction affects temporal leakage. Here, the number of taps per channel is two. The subplots in this figure display the following:

1. The original input signal, a time domain impulse of width one.

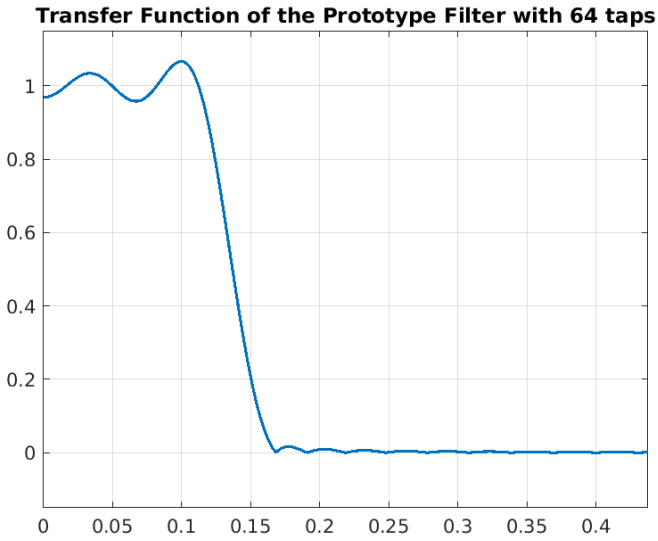

Figure 5: 64 tap FIR filter response. Note the ripple in the passband

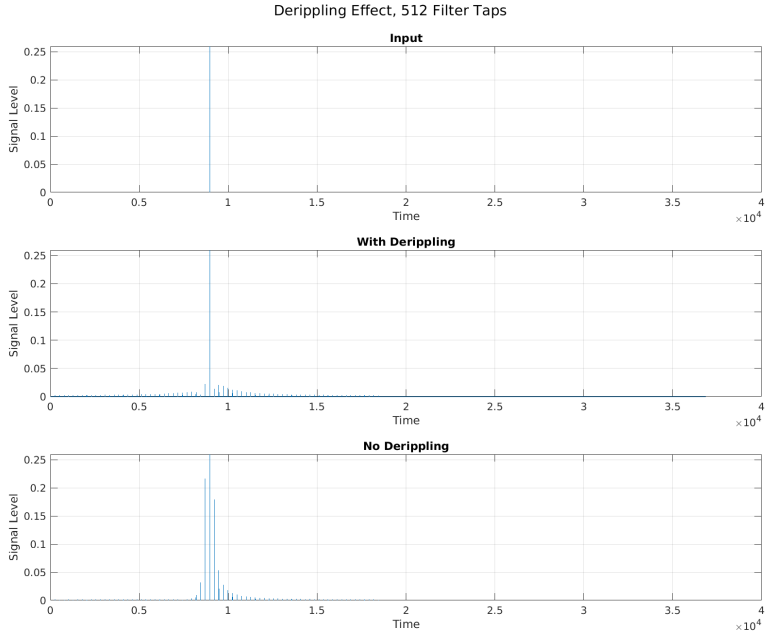

Figure 6: Effect of derippling correction on temporal leakage

2. The result of PFB inversion, with derippling enabled.
3. The result of PFB inversion, with derippling disabled.

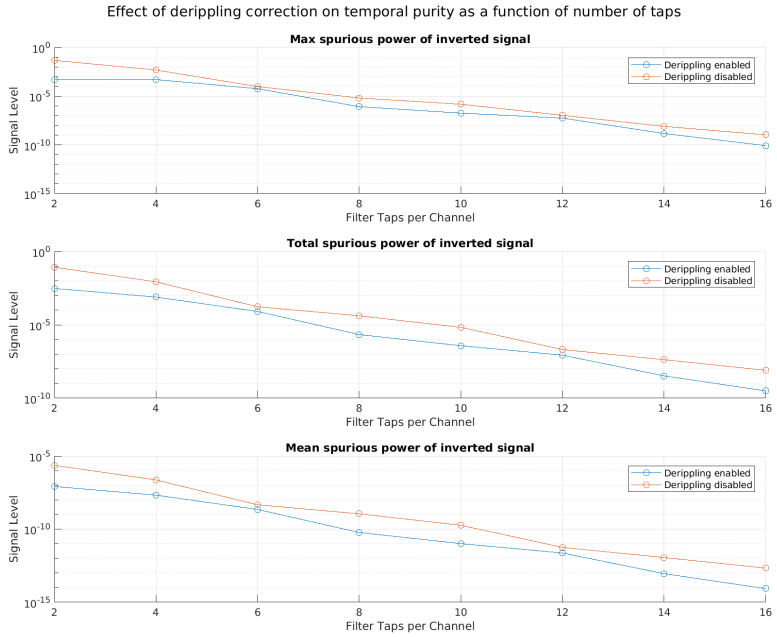

Figure 7: Effect of derippling on temporal purity measures as a function of the number of filter taps per channel

Figure 7 shows the effect derippling can have measures of temporal purity as a function of the number of filter taps per channel. Here, I use the max, mean, and total spurious power of time series as measures of temporal purity. Figure 7 contains three subplots, each showing one of these purity measures. Each subplot has two lines. The blue line shows the purity measure with derippling enabled, while the red line shows the same measure with derippling disabled, all as a function of number of filter taps.

Derippling results in decreased temporal leakage. The effect is more pronounced for a smaller number of filter taps. For larger number of taps, the passband of the FIR response becomes flat enough that derippling is no longer effective.

Derippling and spectral purity

Figure 8 shows how applying the derippling correction affects spectral leakage. Like with the time domain example above, the number of taps per channel is two. The subplots in this figure display the following:

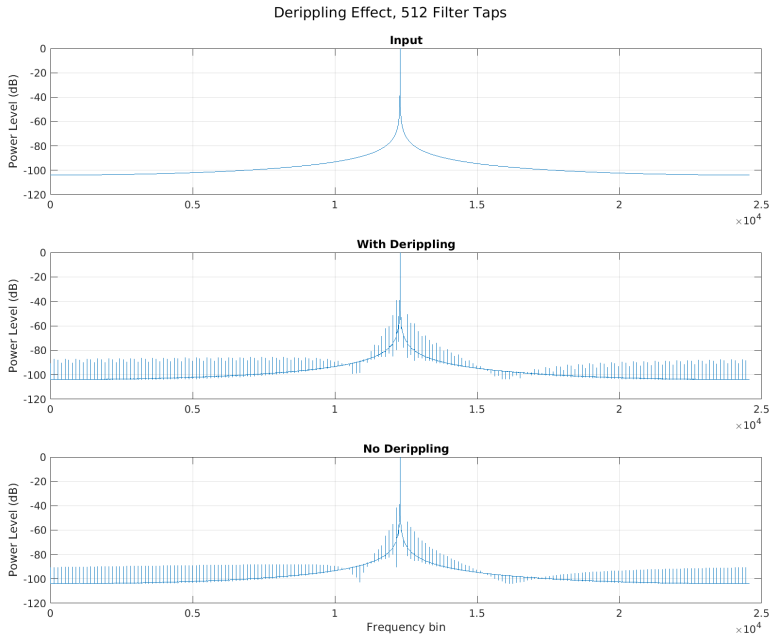

Figure 8: Effect of derippling correction on spectral leakage

1. The power spectrum of the original input signal, a low frequency complex sinusoid. Here, the sinusoid's frequency is not bin-centred.
2. The power spectrum of the result of the PFB inversion, with derippling enabled.
3. The power spectrum of the result of the PFB inversion, with derippling disabled.

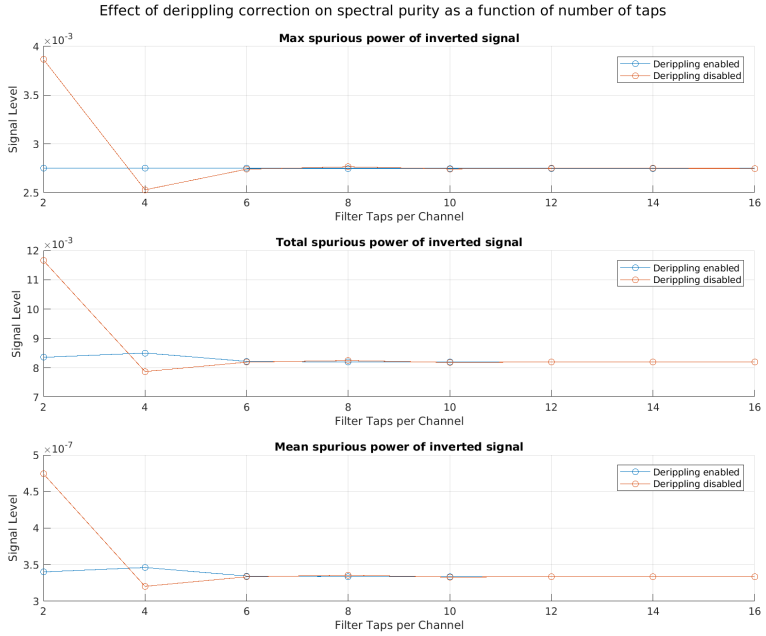

Figure 9: Effect of derippling on spectral purity measures as a function of the number of filter taps per channel

Figure 9 shows the effect of derippling on measures of spectral purity as a function of the number of filter taps per channel. I use the max, total and mean spurious power of power spectra as measures of spectral purity.

The relationship between spectral purity and derippling is not as clear cut as with temporal purity. We don't reap the same benefits from derippling as in the time domain; in fact, derippling can even have a negative effect on spectral purity, as seen with the four taps per channel case in figure 9.

Overlap discard

When processing more than one block of data, it becomes necessary to deal with the edge effects of the FFT. If adjacent blocks of inverted data are simple

concatenated together significant temporal leakage can occur between blocks. Figures 10, 11, 12, and 13 attempt to show temporal leakage caused by this effect. Each figure has two columns, each with three subplots. The first columns shows the original input signal, the inverted signal and the numerical difference between the two in the time domain. The second column shows the same in the frequency domain. In figures 12, and 13, the level of the time domain signals is shown in decibels.

Note I used 256 channels with 4/3 oversampling for the plots in this section. The FIR filter had 10 taps per channel. The input sinusoid has a frequency of 4.1 Hz, meaning that it is not bin-centred in the frequency regime. Every time domain impulse presented is of width one.

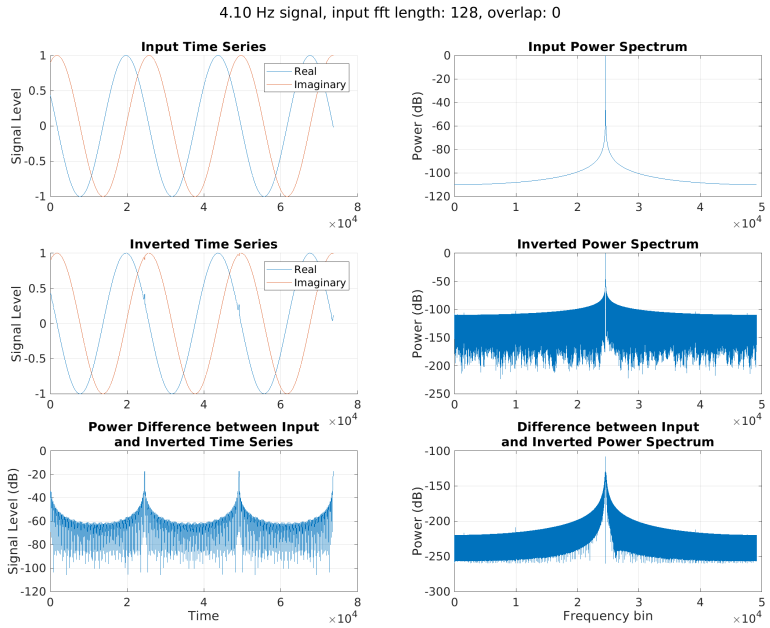

Figure 10: No overlap discard. Notice significant spikes in difference between time domain signals.

Figure 10 shows the result of not using any overlap discard technique. There is significant temporal leakage at the boundaries between the processing blocks, as indicated by the spikes in the difference between the input and inverted time series.

Figure 11 shows the effect of introducing an overlap discard technique in the PFB inversion implementation. This technique rewinds through data that has already been analyzed in addition to discarding data at the beginning and end

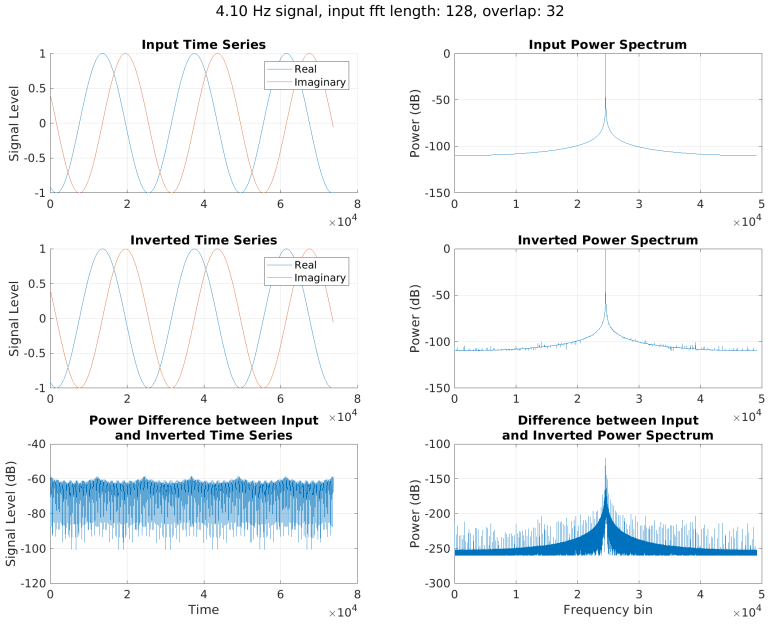

Figure 11: 32 sample overlap region

of computational blocks. This significantly reduces the error at the block edge.

Using time domain impulses as input especially highlights temporal leakage from improperly handled block edges. By placing an impulse on the block edge, we can see how introducing an overlap discard technique reduces, but does not eliminate this source of temporal leakage.

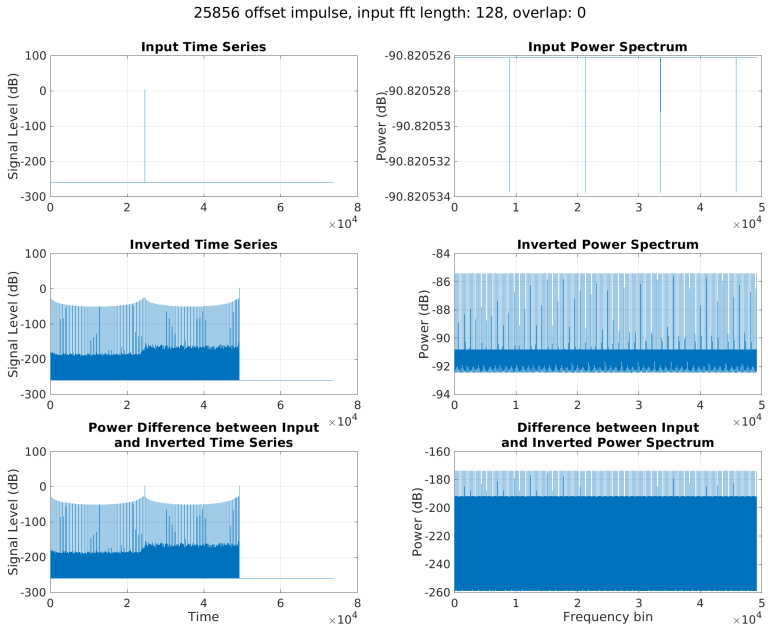

Figure 12: No overlap region.

Figure 12 shows the result of pushing an impulse through the PFB inversion signal chain with no overlap discard technique. In this example, the impulse is placed right on the boundary between the first and second processing block. In the inverted signal, the impulse shows up at the end of the second block, while simultaneously leaking throughout the first and second block.

Figure 13 shows the result of introducing a 32 sample overlap region on either side of adjacent blocks. Here, the temporal leakage is significantly reduced, but not to the level of what we would expect to say if the impulse were placed safely in the middle of a block.

Figure 14 shows measures of temporal purity for a range of impulse positions. *Unlike previous examples, the number of channels being inverted is 16.* The reason for using a reduced number of channels is computational expediency. Here, the overlap is 64 samples. This plot demonstrates that placing impulses

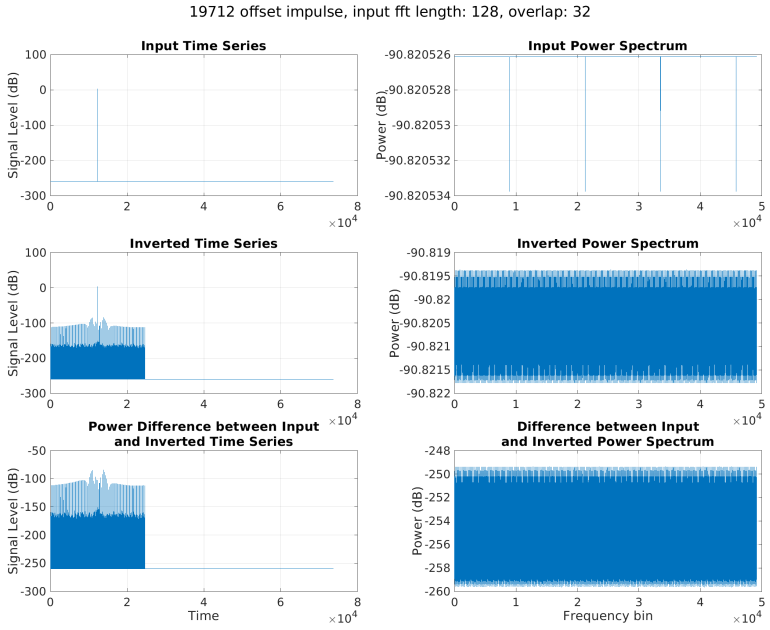

Figure 13: 32 sample overlap region. Notice the significant attenuation of temporal leakage

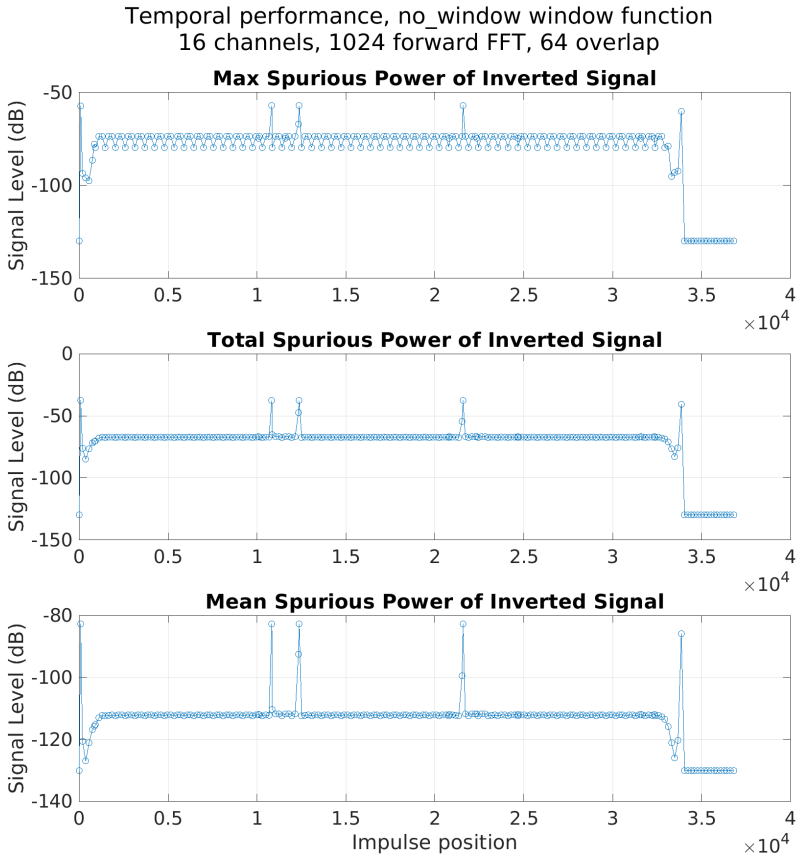

Figure 14: Temporal purity with no window function, using 32 sample overlap. Note the jumps in error corresponding to block edges.

at the boundary between processing blocks introduces a significant increase in error.

Overlap Region and FFT window functions

The overlap discard method improves block edge induced temporal leakage effects, but it does not ammeliorate it enough to meet the PST specification. In order to drive down time domain error, we investigated a number of FFT window functions with the intent of smoothing or overstepping sharp edges in data introduced to the FFT. We apply these window functions before the forward FFT step of the PFB inversion algoritm. In other words, we apply window functions to blocks of channelized and downsampled time series data.

In particular, we looked at the Hann, top hat, "fedora" and Tukey windows. Figure 15 shows these window functions. Here, the fedora window is a modified top hot in which the acceptance region reaches into the discard region. Of the windows we investigated, the Tukey window performed best for the widest range of FFT lengths and input overlap sizes. Indeed, we found that the fedora window was able to stop temporal leakage, but only for high FFT lengths and large discard regions.

Figure 16 shows the performance of the Tukey window for a variety of time domain offsets. Here, the forward FFT length is 1024, the overlap region is 64 samples *and the number of channels is again 16*.

After probing the FFT length and overlap region parameter space, we found that the required overlap size appears to be constant (as opposed to some fractional amount of the FFT length), provided the forward FFT is long enough. The size of this optimal overlap is dependent on the number of the channels the upstream PFB is producing. Generally speaking, the greater the number of channels, the greater the size of the forward FFT required, and the greater the overlap size.

Overall PFB inversion performance

With the introduction of derippling, overlap discard with FFT windows, and proper phase alignment, the FFT based PFB inversion algorithm is well poised to perform within the PST specification.

From the PST signal model document, the recommended levels for error measures are as follows.

1. max-max-spurious power – temporal (-70 dB)
2. max-max-spurious power – spectral (-70 dB)
3. max-total-spurious power – temporal (-60 dB)
4. max-total-spurious power – spectral (-60 dB)

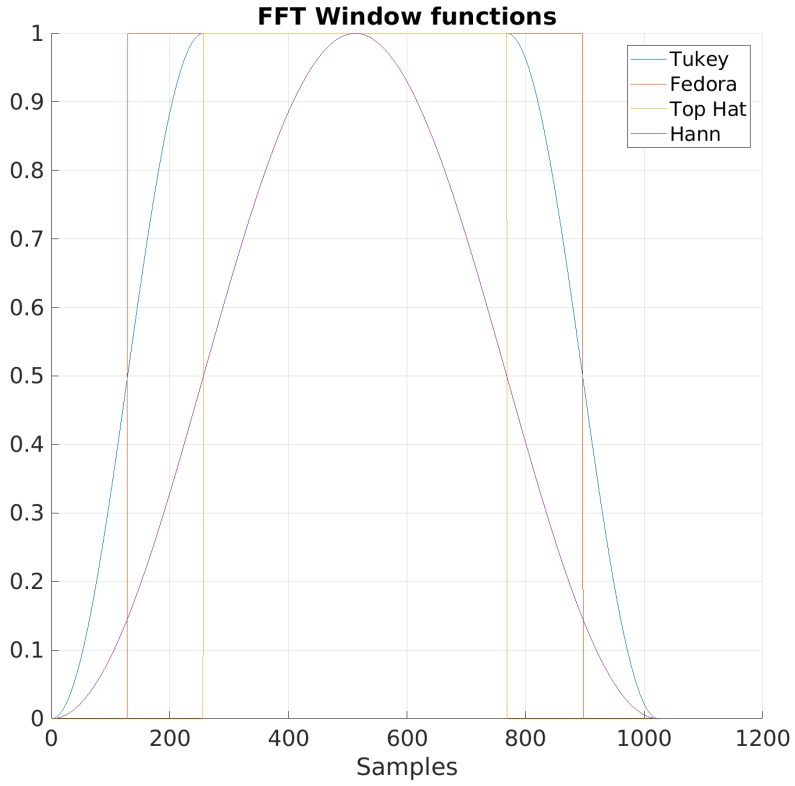

Figure 15: FFT window functions. Here the number of samples is 1024, and the overlap region is 128 samples at the beginning and end.

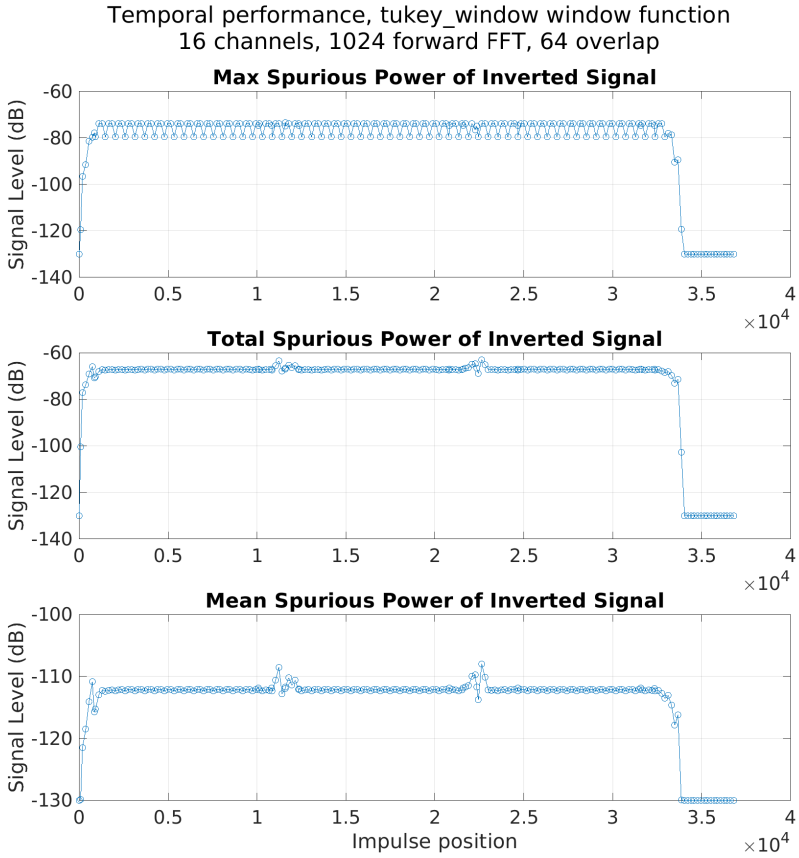

Figure 16: Temporal purity with the Tukey FFT window, for 16 channels.

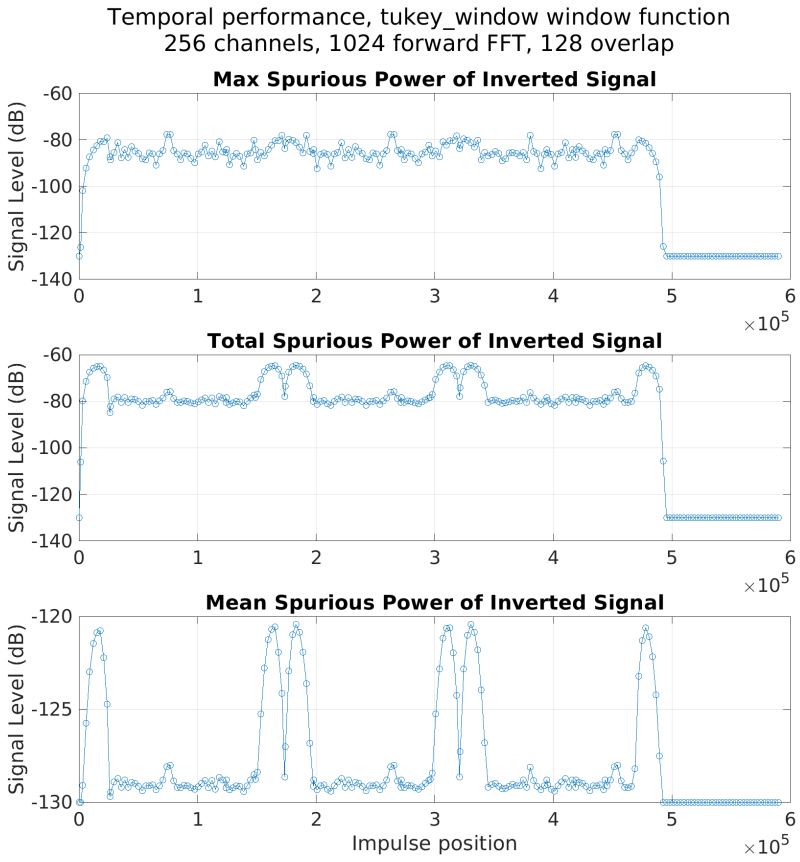

Figure 17: Temporal purity with the Tukey FFT window, for 256 channels.

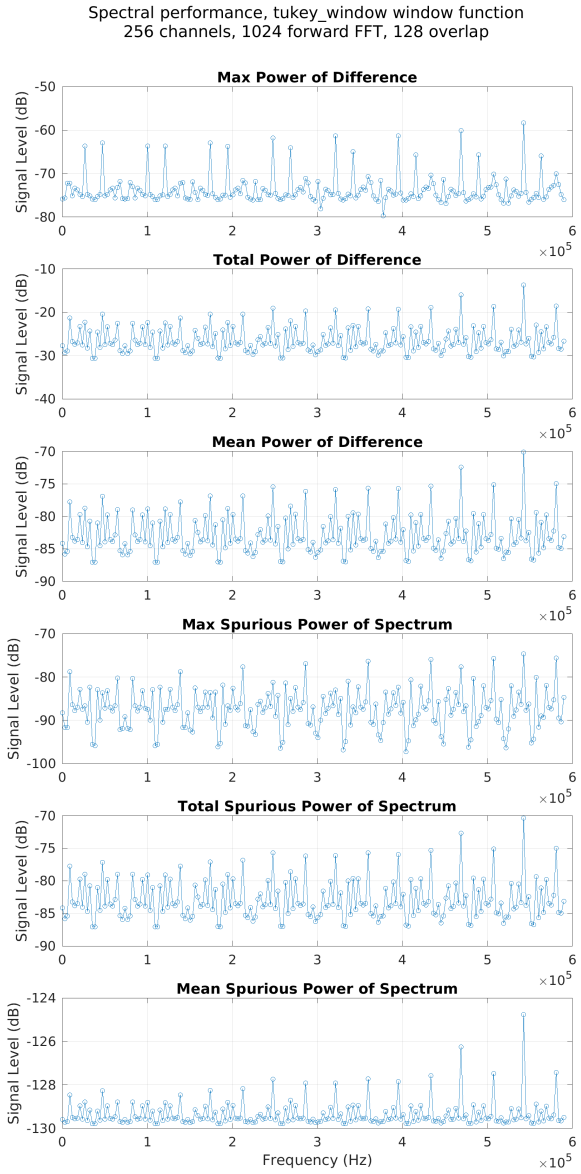

Figure 18: Spectral purity with the Tukey FFT window, for 256 channels.

I'll address each of these measures individually, using figures 17 and 18 as visual references. These figures use 256 channels, 1024 sample forward FFT, with 128 sample overlap region. Better performance can be acheived using larger FFT lengths.

Figure 18 shows the performance of the Tukey window for complex sinusoids of a range of frequencies. The first three subplots show the max, total and mean power of the complex difference (in the time domain) between input and inverted signals. The last three subplots show the max, total and mean spurious power of the spectrum of the inverted data. The purpose here is to illustrate that good spectral purity also results in good numerical reconstruction of signals.

1. The temporal purity is below -70 dB. Close to the block edges, the error does creep up close to the reference measure.
2. For the bin-centred case, the spectral performance is within specification.
3. For non block edge impulses, the total spurious power is well below -60 dB. It does start to come close to the limit near block edges.
4. As with the max spurious power case, the spectral performance is within the recommended specification.